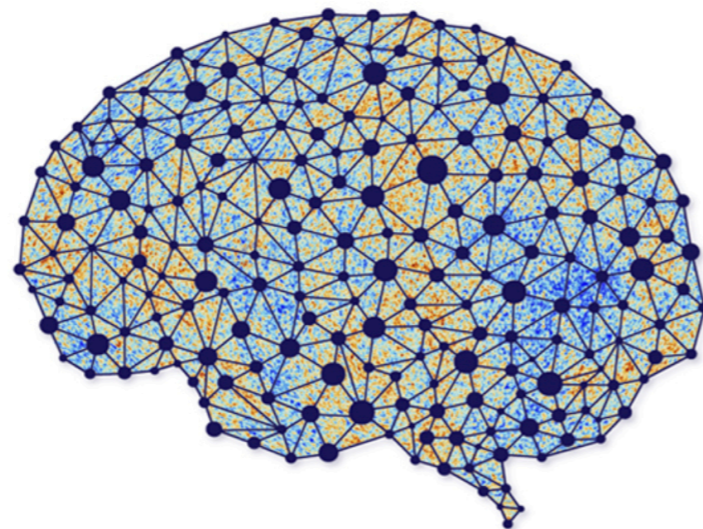


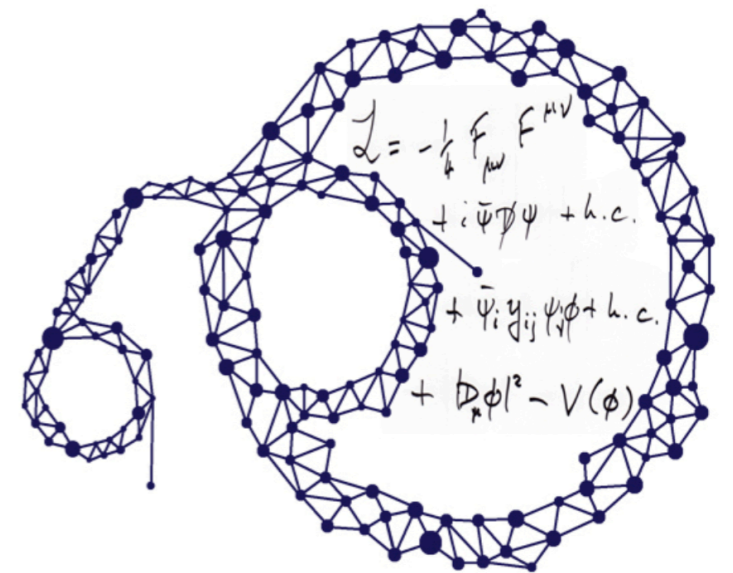
Physics 361 - Machine Learning in Physics

Lecture 18 – Random Forests with Scikit-learn, Final project

Mar. 21st 2024



AI
∩
Universe



Moritz Münchmeyer

Decision tree ensembles

4. Details of the training process

Some slides and plots from: Viviana Acquaviva - Machine Learning for physics and Astronomy

In our decision tree training notebook we will encounter a few concepts which we have not yet discussed in the lecture.

These are not specific to decision trees, but let's cover them here. In particular we need to know about

- R^2 -score
- Cross-Validation
- Hyperparameter optimization

R²-score

- The tree-regressor in scikit learn does not report the MSE loss, but rather the R² score. How does it measure the quality of a regression?
- How “strong” is relationship between predictors & outcome?
- We can attempt to measure the fraction of observed variance of the target variable (“outcome”) that can be explained by the features (“predictors”):

$$SS_{\text{tot}} = \sum_{i=1}^N (y_i - \bar{y})^2; \quad SS_{\text{res}} = \sum_{i=1}^N (y_i - \hat{y}_i)^2,$$

- where y_i = true values,
 - the bar denotes the mean, and the hat denotes the prediction.
- SS_{tot} = Sum of Total Squares
 - SS_{res} = Sum of Squares due to Residuals

R²-score

How “strong” is relationship between predictors & outcome?

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

where:

SS_{res} = sum of squares due to residuals

SS_{tot} = total sum of squares

1 indicates perfect correlation, and 0 indicates no relationship (a model that predicts the mean of true values will have $R^2 = 0$).

R² can be negative on the test set (e.g., the predictions can be arbitrarily bad, worse than the mean!)

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \in (-\infty, 1]$$

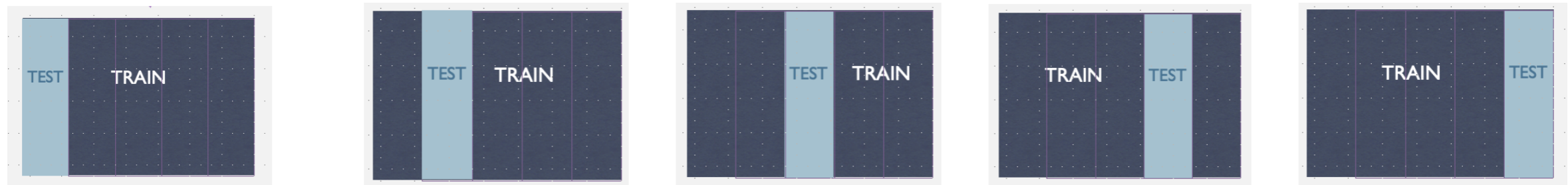
R²-score

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

- Interpretation: If R² has a value of 0.6, this means 60% of the variation in the dependent variable (y) is explained by your regression model. The remaining 40% is unexplained.
- Be careful with interpretation:
 - The higher the value of R², the better the model fits the data.
 - When it is used in a machine learning setting (i.e. we look at R² for predictions, on test set), it is a useful tool to compare models.
 - If features and targets are the same, a model with higher R² score on the predicted values is better.
 - But we can't distinguish between a poor R² that comes from bad modeling, and a poor R² that comes from noisy data.

Cross-Validation

- We have not yet explained an important practice of supervised training, of particular importance when the data set is small: Cross-Validation
- The idea is to split the data set in multiple ways into training and test data. Then we train the model K times, and evaluate its average performance.
- This is a good idea because:
 - We want to use all the data for training (and not “lose” the test data)
 - We avoid the risk of under/overestimating performance because of a non-typical performance of a particular training/test split.
 - We get an estimate of how much scores fluctuate because of variance in the data.
- K-fold cross validation looks like this (for K=5)



- Disadvantage: We need to train the model K times, which takes more computation time.

Hyperparameter tuning

- The most common procedure to optimize hyperparameters is a cross-validated **Grid Search of the hyperparameter space**.
- As a reminder, we do this to pick optimal hyperparameters, but the test scores we obtain are still optimistic (there is leakage of information between the optimization and the test scores).
- The correct procedure involves a 3-tiered structure: train/validation/test
- Like with any parameter space search, the grid method can be inefficient, or too time-consuming. Alternatives are varying parameters one at a time (ignores correlations), Random Search (often good enough), Bayesian parameter search.
- There are elaborate hyper parameter optimization algorithms. Libraries specifically designed for this purpose include:
 - RayTune <https://docs.ray.io/en/latest/tune/index.html>
 - HyperOpt <https://github.com/hyperopt/hyperopt>
 - These libraries include several hyperparameter tuning algorithms. Rather than just performing a grid search, these algorithms optimize the hyper parameters by approximating the gradient of the optimization target (e.g. MSE) with respect to the hyper parameters.

Hyperparameter tuning with scikit-learn

- It is recommended to search the hyper-parameter space for the best cross validation score.
- A search consists of:
 - an estimator (regressor or classifier such as `sklearn.tree.DecisionTreeRegressor`);
 - a parameter space;
 - a method for searching or sampling candidates;
 - a cross-validation scheme; and
 - a score function (e.g. R2 score).
- Two generic approaches to parameter search are provided in scikit-learn: for given values,
 - `GridSearchCV` exhaustively considers all parameter combinations while
 - `RandomizedSearchCV` can sample a given number of candidates from a parameter space with a specified distribution.
- We will use `GridSearchCV`. It systematically works through multiple combinations of parameter tunes, cross-validating as it goes to determine which tune gives the best performance based on a specified score.
 - https://scikit-learn.org/stable/modules/grid_search.html#grid-search

Decision tree ensembles

5. Application: Red Shift estimation (cont.)

Slides and python notebook from: Viviana Acquaviva - Machine Learning for physics and Astronomy chapter 6

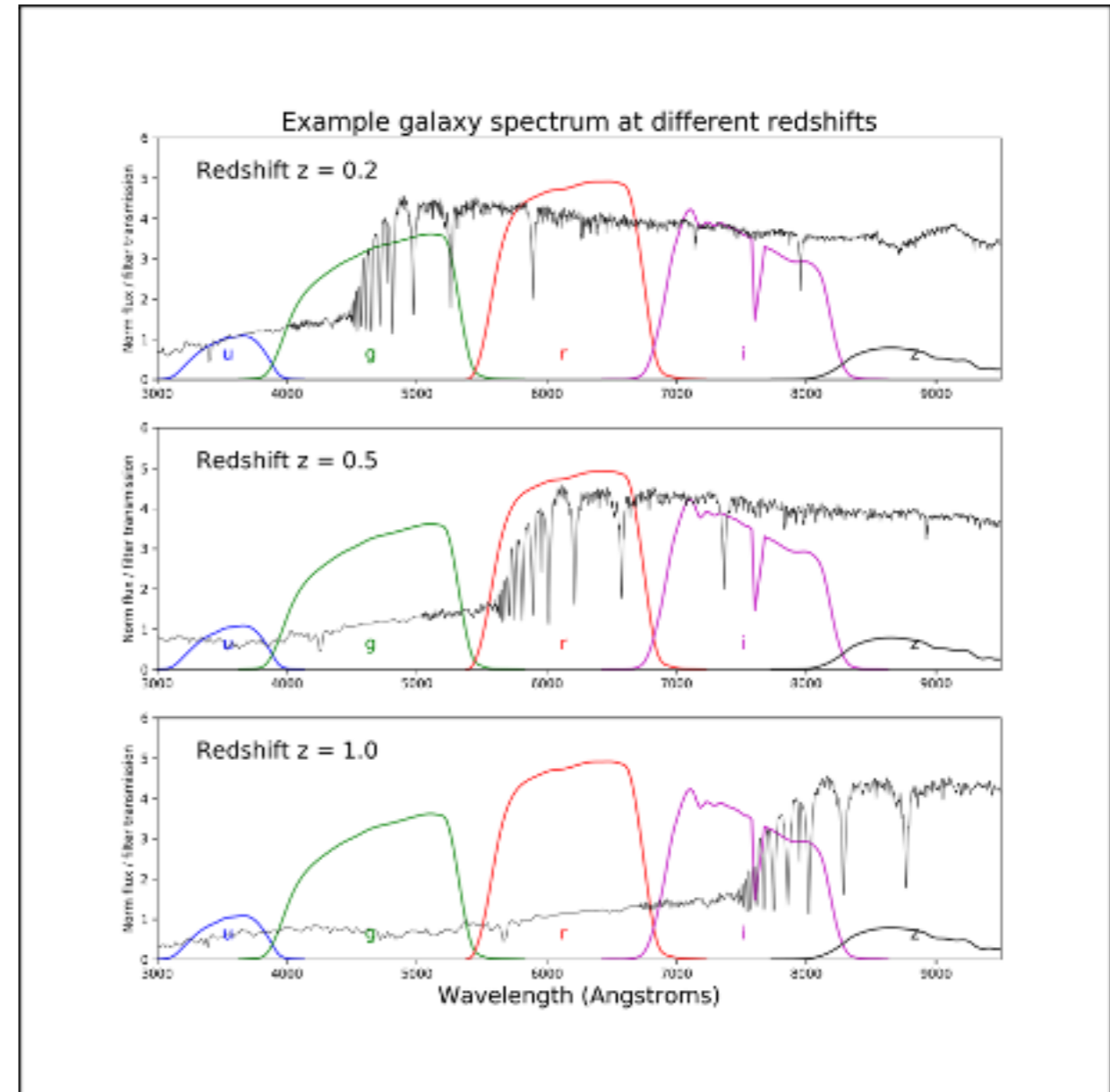
Learning task

Input data:

Collection of photometric intensity in 6 bands (i.e. 6 numbers per galaxy)

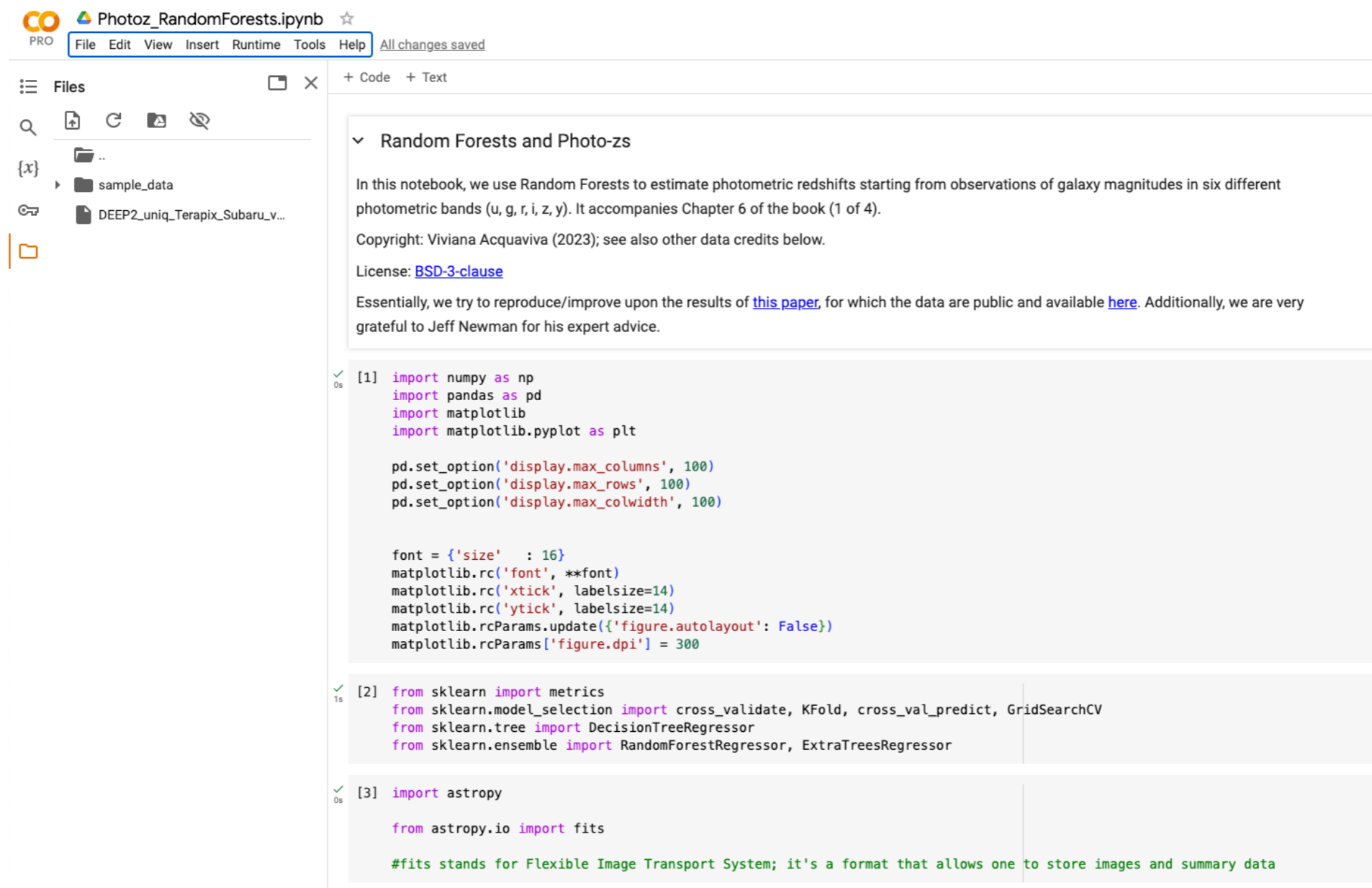
Target data:

True redshift of the galaxy obtained from more expensive spectroscopy. 1 number called z .



Colab notebook

The rest of this lecture will be on Colab. We will use notebooks from the book Viviana Acquaviva “Machine learning for Physics and Astronomy”. The notebooks can be downloaded on the course website, and on the book website <https://press.princeton.edu/books/paperback/9780691206417/machine-learning-for-physics-and-astronomy>.



Photoz_RandomForests.ipynb ☆
File Edit View Insert Runtime Tools Help All changes saved

Files

- sample_data
- DEEP2_uniq_Terapix_Subaru_v...

Random Forests and Photo-zs

In this notebook, we use Random Forests to estimate photometric redshifts starting from observations of galaxy magnitudes in six different photometric bands (u, g, r, i, z, y). It accompanies Chapter 6 of the book (1 of 4).

Copyright: Viviana Acquaviva (2023); see also other data credits below.

License: [BSD-3-clause](#)

Essentially, we try to reproduce/improve upon the results of [this paper](#), for which the data are public and available [here](#). Additionally, we are very grateful to Jeff Newman for his expert advice.

```
[1] import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt

pd.set_option('display.max_columns', 100)
pd.set_option('display.max_rows', 100)
pd.set_option('display.max_colwidth', 100)

font = {'size' : 16}
matplotlib.rc('font', **font)
matplotlib.rc('xtick', labels=14)
matplotlib.rc('ytick', labels=14)
matplotlib.rcParams.update({'figure.autolayout': False})
matplotlib.rcParams['figure.dpi'] = 300
```

```
[2] from sklearn import metrics
from sklearn.model_selection import cross_validate, KFold, cross_val_predict, GridSearchCV
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, ExtraTreesRegressor
```

```
[3] import astropy

from astropy.io import fits

#fits stands for Flexible Image Transport System; it's a format that allows one to store images and summary data
```

Final project

Final project

- You will write a **paper on an application of machine learning to physics** of your choice. Your paper needs to contain a computational analysis, which generally will mean applying a machine learning method to some data set.
- You can **work alone or in groups of two**.
- The paper should be **5 to 10 pages** and contain the following:
 - A short review of at least one research paper related to your topic. This is to encourage you to learn how to browse the literature.
 - A description of the data set you will be working with and its properties.
 - A brief description of the machine learning method you will use. Don't re-explain basics such as how CNNs work, rather describe the detailed properties of your approach.
 - Train the model and put the results in your paper. Explore some variations such as different hyper parameters.
 - Describe successes and problems in your analysis.
 - Also **submit the Colab notebook** you used for training/evaluation.
- If you are already doing research in physics or a related field, you can write the paper on this topic if you wish.

Final project

- You can **use machine learning methods either from the lecture or ones that we have not covered**. Major topics which we have not yet covered but will be covering in the next weeks are Generative models (GANs, Diffusion, Normalizing Flows) and Simulation Based Inferences (which includes the important topic of assigning error bars to neural network estimates).
- The project should take you ~three days of work, spread over the last weeks of the semester.
- Your **paper will be due on Sunday May 5th at midnight**.
- We want to **know your topic by April 16th at latest**. You can discuss your topic ideas with Gary after his lectures on April 2, 4, 9,11 or with Moritz after the lecture on April 16. You can also schedule an appointment.

Some sources of data

- Kaggle competitions
 - <https://www.kaggle.com/search?q=physics>
 - <https://www.kaggle.com/search?q=physics+in%3Adatasets>
 - <https://www.kaggle.com/search?q=physics+in%3Acompetitions>
 - <https://www.kaggle.com/competitions/icecube-neutrinos-in-deep-ice>
- Shared Data and Algorithms for Deep Learning in Fundamental Physics
<https://github.com/erum-data-idt/pd4ml>
- https://github.com/drckf/mlreview_notebooks/tree/master/jupyter_notebooks/notebooks
- <https://astronn.readthedocs.io/en/latest/galaxy10.html>

Some sources of models

- Scikit learn: <https://scikit-learn.org/stable/index.html> (basics, but important)
- <https://pytorch.org/tutorials/> (includes things like GANs)
- <https://docs.pyro.ai/en/stable/> (probabilistic machine learning framework)
- <https://sbi-dev.github.io/sbi/> (library for simulation-based inference)
- Many git repositories associated with papers
- I would perform a web search on machine learning papers on a physics topic that you are interested in and get inspired by their data and model. Of course you need to simplify your approach substantially compared to a research paper.

Course logistics

- **Reading for this lecture:**
 - This lecture was based mostly on Viviana Acquaviva “Machine learning for Physics and Astronomy” chapter 6.